REMARKS

This is intended as a full and complete response to the Final Office Action dated March 25, 2005, having a shortened statutory period for response set to expire on June 25, 2005. Please reconsider the claims pending in the application for reasons discussed below.

Claims 1-29 are pending in the application. Claims 1, 3-15, 17-21 and 23-29 remain pending following entry of this response. Claims 1, 3-5, 14, 17, 21, 23-26 and 28-29 have been amended. Claims 2, 16 and 22 have been cancelled.

Claim Rejections - 35 *USC* § *102*

Claims 1, 3-15, 17, 21, 23-29 are rejected under 35 U.S.C. 102(e) as being anticipated by *Ungar* (US. 6,085,035).

Applicants respectfully traverse this rejection. "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990). *Ungar*, however, fails to disclose selecting a call linkage from at least one of a memory-based call linkage and a register-based call linkage.

Regarding claims 1, 14, 21, and 29, the Examiner argues that *Ungar* discloses: selecting a call linkage between a caller procedure and a callee procedure for each procedure call using the extracted information, where the selected call linkage is optimized to minimize the run time of the object code generated from the source code; generating the object code from the source code; and running the object code using the selected call linkages for each procedure call. Specifically, the Examiner asserts that this limitation is disclosed by the following passage from *Ungar*:

> A first preferred embodiment optimizes both a called routine and the call site dependent on the types of the data-value passed from the call site to the called routine. Because data-values contained in passed entities (that is, the data-values contained in variables and/or the addresses of the variables themselves) can be specified as arguments to, or a result from, a called routine, the call site generally contains code to select which executable version of the called routine to invoke dependent on the types of the passed entities. The invention detects variables that have immutable types (from the 'determine type usage pattern' procedure 305) and optimizes both the called routine and the call site dependent upon the type-mutability of the passed entities. Additionally, if the variables have mutable types, the invention generates multiple versions of the called routine (each optimized for a preferred type as determined by the 'determine type usage pattern' procedure 305) that are invoked dependent on the types of the passed data-values. Often, one of these called routine versions is not optimized with respect to any of the passed data-values and so is capable of processing any pattern of types of the passed data-value.

Ungar 8:52-67 – 9:1-7. The material cited by the Examiner shows that *Ungar* discloses techniques to optimize "both a called routine and the call site dependent on the types of the data-value passed from the call site to the called routine." *Ungar*, 8:52-54. *Ungar* defines "data value and data type as follows:

> Data type—The type of the data-value. The data type is either associated with the data-value itself, or with the variable that contains the data-value. There are primary types (for example integer and real) and constructed types (such as those defined by data structures.) Some variables store data-values of only one type.
>
> Data-value—The data-value is a pattern of bits that have a meaning that depends on the data type associated with the data-value.

Ungar, 5:1-22. Thus, *Ungar* discloses optimizing a compiled procedure based on data types associated program variables used to store data-values. Further, variables (and associated data-types) may be characterized "immutable" (i.e. non-changing) "preferred" (i.e. likely to be a certain type) or "mutable" (i.e. may be of any type). As described in paragraph 31 of the application, however, "a call linkage represents a link or relationship between a procedure call (caller procedure) and a procedure that is called (callee procedure). In one embodiment, the compiler 110 selects between a memory-based call linkage and a register-based call linkage. ... In the memory-based

call linkage, the parameters or arguments of a procedure or subroutine call are "passed in memory." Namely, the arguments are initially stored from the registers 128_N to memory 106 and then loaded from memory 106 back to the registers 128_N ... In the register-based call linkage, the arguments of the procedure are copied to and from registers in the processor 102. A particular subset of registers 128_N, called parameter registers, are used during a procedure call." *Application*, ¶ 31 (emphasis added).

In short, selecting a call linkage determines the mechanism used to pass parameters to a procedure: Memory based call linkage stores the values in memory, register based call linkage stores this data in parameter registers. Nothing in the material cited by the Examiner discloses selecting a call linkage from one of a memory-based call linkage and a parameter-based call linkage. Rather, it discloses compiling multiple versions of a called procedure with different versions optimized for variables that may be used to store different data-types (e.g., the "mutable type," "immutable type," or "preferred type). *Ungar* is silent, however, regarding any optimizations regarding how the parameters are actually passed to a procedure, (i.e., it is silent on selecting a call linkage).

Respectfully, Applicants believe that the Examiner may have confused the limitation of selecting a call linkage with the techniques for optimizing procedure calls based on differing data types as disclosed by *Ungar*. Accordingly, in order to clarify the term "call-linkage" Applicants have amended Claims 1, 14, and 21, to include the limitation originally specified by claims 2, 16, and 22, specifying that the call linkage is selected from at least one of a memory-based call linkage and a register-based call linkage. Regarding these claims (2, 16, and 22) the Examiner rejected these claims citing material describing *Ungar*, 9:22-67 – 10:1-21. Each of the optimization techniques disclosed by this material, however, determines whether the "data-value" is accessed (e.g., using a procedure call or a reference to a local variable) using a variable that has a "mutable type," an "immutable type," and a "preferred type," and provides optimized routines, accordingly. Thus, *Ungar* fails to disclose selecting a call linkage and further fails to disclose selecting a call linkage from at least one of a memory-based call linkage or a register-based linkage. Applicants respectfully submit,

therefore, that claims 1, 14, 21, and 29 are patentable over *Ungar* and request that the rejection be withdrawn.

Regarding claims 3, 4, 23, and 24, each of these claims include additional limitations depending on whether a "memory based" call linkage is selected (claims 3 and 23) or a "register based" call linkage is selected (claims 4 and 24). The Examiner cites same the material referred to above in rejecting these claims. However, as discussed above, *Ungar* fails to disclose selecting the appropriate "call linkage" from at least one of a memory-based call linkage and a register-based call linkage." Furthermore, claims 3 and 23 include the additional limitation of "copying a value, for each argument in a procedure call, from a register in the processor to a parameter register in the processor" and claims 4 and 24 include the additional limitation of "allocating a block in memory to store a value for each argument in a procedure call." *Ungar* does not disclose the steps for passing parameters to a procedure based on a selected call linkage that includes specifically storing data in parameter registers, or allocating blocks of memory to do the same. Applicants respectfully submit, therefore, that claims 3, 4, 23, and 24 are patentable over *Ungar*, and request that the rejection be withdrawn.

Regarding claims 5, 17 and 25, each of these claims includes a further limitation of "detecting whether an error exists for the procedure call." In rejecting these claims, the Examiner cites to the portion of the material cited in the rejection of claims 2, 16, and 22. As described above, however, this material is directed to evaluating data-types in procedure calls based on a "mutable type," an "immutable type," and a "preferred type" characteristic to optimize a compiled versions of the procedure call. Moreover, *Ungar* does not disclose detecting whether an error exists for the procedure call, nor does *Ungar* disclose selecting a call linkage based on the presence of a detected error. Simply, *Ugnar* does not disclose how to manage any error conditions that may occur at all. Applicants respectfully submit, therefore, that claims 5, 17, and 25 are patentable over *Ungar*, and request that the rejection be withdrawn.

Regarding claims 6-13, 15, 18-20, 26-28, each of these claims depends from one of claims 1, 14, 21. For the reasons set forth regarding the these independent claims, Applicants respectfully submit that claims 6-13, 15, 18-20, 26 and 27 are patentable over *Ungar*, without the need for further remarks by Applicants, and request that the rejection be withdrawn.

Conclusion

Having addressed all issues set out in the office action, Applicant respectfully submits that the claims are in condition for allowance and respectfully request that the claims be allowed.

Respectfully submitted,

Gero G. McClellan
Registration No. 44,227
MOSER, PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicant(s)